(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
17 May 2001 (17.05.2001)

PCT

(10) International Publication Number
WO 01/35571 A1

(51) International Patent Classification[7]: H04L 9/00

(21) International Application Number: PCT/US00/30899

(22) International Filing Date:
9 November 2000 (09.11.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/436,916 9 November 1999 (09.11.1999) US

(71) Applicant: WIDEVINE TECHNOLOGIES [US/US]; 1301 Fifth Avenue, Suite 1300, Seattle, WA 98101 (US).

(72) Inventor: BAKER, Brian; 14631 NE Third, No. 7, Bellevue, WA 98007 (US).

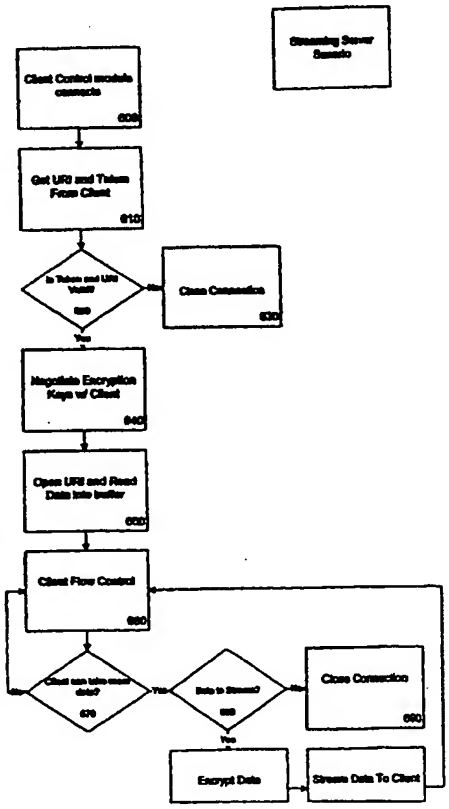
(74) Agent: BRUESS, Steven, C.; Merchant & Gould, P.C., P.O. Box 2903, Minneapolis, MN 55402-0903 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: PROCESS AND STREAMING SERVER FOR ENCRYPTING A DATA STREAM



(57) Abstract: There is disclosed a process for encrypting a data stream (700) to secure the data stream for single viewing and to protect copyrights of the data stream. Specifically, there is disclosed a process for protecting streaming multimedia, entertainment and communications in an Internet-type transmission. There is further disclosed a streaming server component operably connected with a streaming server that interacts with a client system (710) to effect the inventive process.

WO 01/35571 A1

Published:

— *With international search report.*

— *Before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

PROCESS AND STREAMING SERVER FOR ENCRYPTING A DATA STREAM

Technical Field of the Invention

The present invention provides a process for encrypting a data stream to secure the data stream for single viewing and to protect copyrights of the data stream. Specifically, the invention provides a process for protecting streaming multimedia, entertainment and communications in an Internet-type transmission. The invention further provides a streaming server component operably connected with a streaming server that interacts with a client system to effect the inventive process.

Background of the Invention

The Internet has provided another means for communication whereby data can be streamed from a server to a client. The client is responsible for displaying the streamed data, preferably streamed media, to a user. The server is responsible for delivering the data stream to the client. The Real Networks and Microsoft solutions send the data stream via a UDP (a connectionless Internet protocol) along with another connection between the client and the server that controls the transmission of the streamed data. The control connection element functions to stop buffer overruns and can adjust the transmission of the stream to compensate for bandwidth latencies. One problem with this arrangement, however, is that the data that are streamed to the client from the server are unprotected and available to anyone on the network. Therefore, there is a need in the art to better protect from interception across a wide area network, such as the Internet. Specifically, the need relates to providing an ability to protect the improper interception and ability to copy streaming data across the Internet. At present, there is no protection mechanism in place to protect copyrighted data.

Once the data has been release by the server and either received by the user or intercepted before being received by the user, there is no way to restrict the re-transmission of such data once it has been released over a network. Even if the data stream has been copyrighted, there is no means to protect or enforce copyright protection of streamed data. The entity owning the copyright and streaming such content realize that there is no control over what is done with such content after it is released. Therefore, there is a need in the art to provide a means for protecting copyrights in content once streamed over a network. The present invention was designed to address both needs.

Currently, no streaming media solution actually encrypts the data that is being sent from the server to the client. One solution can accomplish this with existing technology, such as by merging SSL secure HTTP sockets with a streaming software package, such as Quicktime. Unfortunately, Quicktime does not have a full screen view option. Therefore, there is a need in the art to develop a better method for streaming video data.

Summary of the Invention

The present invention provides a process for encrypting a data stream to secure the data stream to enable only single viewing, comprising:

(a) providing a client selection for a streaming data transmission;

(b) opening a connection to a streaming server and sending URI, token and user information to the streaming server, wherein the streaming server comprises a client data connection module to send data packets to a client, and encryption module to use encryption keys negotiated with the client to encrypt the data stream and operably connected to the client data connection module, and a flow control module for controlling the rate of data stream flow to maintain a full client buffer;

(c) approving or disapproving a valid or invalid, respectively, URI and token combination on a transaction server, wherein the transaction server comprises a client interaction module for connecting a user to the transaction server component, a user verification module having a user database wherein the user verification module is operably linked to the client interaction module and checking for a valid user, and a URI and token creation module operably linked to the user verification module for creating new URIs and tokens in response to user requests; and

(d) providing a continuously encrypted data stream to the client if a valid URI and token combination was found.

Preferably, the streaming server component further comprises a read buffer module operable connected with the flow control module for reading in data from a source footage on storage medium. Preferably, the streaming server component further comprises a user interface module operably connected to the file system module or flow control module for setting server options. Preferably, the streaming server further comprises a client server component comprising a data stream control protocol module to create an initial connection to the streaming server component, a decryption module to decrypt the incoming data stream, an input buffer module to buffer incoming data streams, and a display control module to control the display of streaming data. Most preferably, the client server component further comprises a display module to display audio and video data.

Preferably, the providing the continuously encrypted data stream step (d) further comprises a user interface module in the streaming server to allow for pausing, stopping, playing or restarting the data stream. Preferably, the transaction server is implemented with ASP scripts for encryption.

The present invention further comprises a streaming server for encrypting a data stream to secure the data stream to enable only single viewing, comprising:

(a) a streaming server component, wherein the streaming server component comprises a client data connection module to send data packets to a client, and encryption module to use encryption keys negotiated with the client to encrypt the data stream and operably connected to

the client data connection module, and a flow control module for controlling the rate of data stream flow to maintain a full client buffer; and

(b) a transaction server component, wherein the transaction server component comprises a client interaction module for connecting a user to the transaction server component, a user verification module having a user database wherein the user verification module is operably linked to the client interaction module and checking for a valid user, and a URI and token creation module operably linked to the user verification module for creating new URIs and tokens in response to user requests.

Preferably, the streaming server component further comprises a read buffer module operable connected with the flow control module for reading in data from a source footage on storage medium. Preferably, the streaming server component further comprises a user interface module operably connected to the file system module or flow control module for setting server options. Preferably, the streaming server further comprises a client server component comprising a data stream control protocol module to create an initial connection to the streaming server component, a decryption module to decrypt the incoming data stream, an input buffer module to buffer incoming data streams, and a display control module to control the display of streaming data. Most preferably, the client server component further comprises a display module to display audio and video data.

Brief Description of the Drawings

Figure 1 shows a schematic of the client component of the enabled to receive and view an encrypted data stream. The client component contains a token storage module 100, a stream control protocol module 120, and a decryption module 160.

Figure 2 shows a schematic of the streaming server component having an encryption module 220 and a client control connection module for key negotiation and token verification 200.

Figure 3 shows a schematic of the transaction server components having a token creation module 330 and a user verification module 310.

Figure 4 shows a schematic of various client scenarios showing the need for a token in order to unlock (decrypt) a data stream for viewing.

Figure 5 shows a schematic of the process for the streaming server showing the receipt of a client token triggering a negotiation of encryption keys to allow viewing and receipt of a data stream.

Figure 6 shows a schematic of the transaction server process providing for setting up of client accounts and token creation.

Detailed Description of the Invention

The present invention provides a process to encrypt a data stream, such as multimedia entertainment and communications, via the Internet. The encrypted data stream will allow for

copyrighted materials and multimedia communications (*e.g.*, analyst meetings) on a secure, pay-per-view basis. The data stream cannot be stored on a client machine for future play-back, or re-transmitted. A client, however, can view a data stream as many times as desired within a specified time frame.

A preferred encryption protocol provides, for example, an encryption algorithm of a 192 bit key (*e.g.*, Triple DES), a UDP packet protocol, a RTSP (rfc 2326) packet transmission protocol, an RTP (rfc 1889) packet transmission control protocol, and MPEG1 video storage compression. However, the foregoing example of a preferred encryption protocol will change as such techniques improve with time.

One advantage of the inventive process, using the inventive streaming server and transaction server, is that the client does not really need to possess fully optimized equipment. Only one client will run on any one machine at any one time. The client will need to playback, for example, 30fps 320x240 video and audio back with no jitter. This will require a stream of about 250~300 kpa, a large data buffer (of at least several megabytes), and a 350 Mhz Pentium II processor or greater running Windows 98 or Windows NT.

The server, for example, can be a fully optimized, multi-threaded (thread pool) Windows NT service. Unlike an HTTP server, this allows sessions with clients to be cached and the server will need to maintain state in respects to all clients.

Definitions

The following terms shall be used with the meanings defined herein.

Client shall mean the computer that the data is being sent to.

User shall mean the person executing instructions on the client.

Module shall mean a collection of compiled code designed to perform a specific function.

URI shall mean universal resource identifier, that is, the location on the server of the stream.

Token shall mean a binary piece of information that describes the permissions the user has for a specific stream.

In a preferred embodiment of the inventive process and streaming server, the video will be stored unencrypted on the server machines; the files will only be retrievable through the server software. The inventive server will be responsible for (1) negotiating a set of encryption keys; and (2) encrypting the video data "on the fly" thereby making the data packets that are actually going over the wire useless to any computer other than the intended machine. A preferred encryption standard is TRIPLE-DES with a 168bit key. This form of encryption is not currently exportable outside of the US and Canada and is extremely secure. The server will use UDP for transmission of the data. This protocol uses considerably less network resources than other TCP protocols (http for example).

Client software will be responsible for decrypting the video data and playback. The encryption keys used will be different every time a movie is accessed. Every time the client is

executed, a different encryption key is created so the client cannot play back earlier streams if they were somehow saved to disk.

Flow Charts

With regard to Figure 1, this shows a schematic of the client component of the inventive process and streaming server enabled to receive and view an encrypted data stream. The client keeps a list of all current streams and the corresponding tokens. This information is stored on the token storage module 100. This list will consist of the following three items: (1) the URI, (2) the token for that URI, and (3) the expiration date given by the server. It is not desirable for the client to have any way of determining if the token is valid or not. Because of this, and the need to remove out of date tokens, the server returns the expiration date. This information is used by the client to display information. The expiration date itself never sent back to the server and the server verifies that the token passed is valid. Examples of module devices that can be used as token storage modules include, for example, Random Access Memory, secondary storage (hard disk), and embedded with software providing for token storage inventory and tracking of expiration dates.

The client communicates with a user interface 110. The client will have a standard user interface that will give the appropriate user experience. The interface will have the ability to look through current valid streams or to connect to the server to search for other streams that could be viewed. The client user interface 110 communicates with a local display control module 130 and a stream control protocol module 120. The client has to be able to setup a communications session with the server as well as control the flow of data from the server once the stream is being viewed. The stream control protocol module 120 creates the initial connection by connecting to the server, passing the requested URI, Token, and user information. The stream control protocol module 120 then negotiates a set of encryption keys and controls the flow of data from the server. Examples of stream control protocol module devices 120 within a client component that can be used to negotiate a set of encryption keys and control the flow of data from a server include, for example, Random Access Memory and the network interface card or modem. The software that will be uploaded into this module will monitor the rate of the data being received by sending network statistics to the streaming server. The display control module 130 controls the display of the data, and has the ability to pause, stop, or re-start the video stream. Examples of display control modules suitable for use within the client component include, Random Access Memory and the video card. The software running in this module will convert the data being sent form the server into a format that can be displayed to the user.

The display module 140 displays video and audio data. The input buffer module 150 is a module that contains the stream buffer. The stream buffer contains a circular buffer of decrypted data that the display control modules reads from and the decryption module writes to. Examples of stream buffer module devices that can be used to contain a circular buffer of decrypted data include, for example, Random Access Memory. As packets are being received

from the server, before the data is put into the input buffer, the data within the transport packet is decrypted by a decryption module 160 using the keys negotiated by the stream control protocol module 120. A decryption module is available commercially, for example SSL, DES, and RSA are available and suitable for use as a decryption module. Lastly on the client component sides is a data stream receive module 170. This module handles the reception of the data packets sent by the server. Appropriate module devices that can be used as a data stream receive module within the client component includes, for example, Random Access Memory. The software contained in this module will save the data being received by the client in a format that can be used by subsequent modules.

With regard to Figure 2, the client control connection module 200 will handle control communications between the client and the server. The client and server will negotiate a set of encryption keys. The client will send user information, the URI, and the token to the streaming server via the client control connection module 200. From this module 200, the data that is streamed to the client can be controlled (that is, paused, stopped, or restarted). Hardware devices suitable for use as a client control connection module within the streaming server, include Random Access Memory. Such hardware components allow for the execution of hardware non-specific operations. Such software is either embedded in the client control connection module or uploaded therein. The software functions to create a process wherein the client and server communicate current network conditions and modify the data stream accordingly.

The client data connection module 210 functions to send data packets to the client using a connectionless protocol to reduce server overhead. Hardware devices suitable for use as a client data connection module within the streaming server, include Random Access Memory and Network Interface Cards. Such software is either embedded in the client data connection module or uploaded therein. The software functions to create a process wherein the encrypted data is sent via network packets to the client machine.

The encryption module 220 uses the keys negotiated by the client/server to encrypt the data stream as it is being sent to the client. This allows for "on the fly" encryption and the encryption keys will be unique for all client/server connections. This allows the source footage to be stored unencrypted on the server. Hardware devices suitable for use as an encryption module within the streaming server, include Random Access Memory and proprietary hardware encryption devices. Such hardware components include software that functions that do the actual encryption of the data. Such software is either embedded in the encryption module or uploaded therein. The software functions to create a process wherein the data being sent to the device is encrypted with the keys originally negotiated with the client and the output data is of a format that can only be read after being decrypted by the client.

The flow control module 230 makes sure that the data stream is being sent to the server at the rate in which the client is using the data. The buffer at the client needs to be full at all times but streaming data must also not be overwritten. Thus, the flow control module

communicates with both the encryption module 220 and uses feedback obtained from the client control connection module 200. Hardware devices suitable for use as a flow control module within the streaming server, include Random Access Memory. Such software is either embedded in the flow control module or uploaded therein. The software functions to create a process wherein the flow of data from the server to the client is regulated.

The file system read buffer 240 is for server performance. Small amounts of data read in from the file will be stored in memory instead of having a constant open file on the file system. The file system module 250 is responsible for reading in data from the source footage on the storage medium. The file system module communicates with the client control connection module 200 to open URIs and the user interface module 260 to file path configurations. Hardware devices suitable for use as a file system module within the streaming server, include Random Access Memory. Such hardware components include software that functions to allow the access to data streams. Such software is either embedded in the file system module or uploaded therein. The software functions to create a process wherein the data stored on the secondary storage device can be loaded into Random Access Memory to be delivered to the encryption module.

The streaming server further provides a simple user interface module 260 for setting server options such as which network port to bind to and the location of source footage. Hardware devices suitable for use as a file system module within the streaming server, include Random Access Memory. Such software is either embedded in the file system module or uploaded therein. The software functions to create a process wherein the user of the server software can tell the file system module where to go to find the data streams.

With regard to Figure 3, the transaction server comprises four module components. To access a video stream, the client must first obtain a transaction token. The transaction token is based on a pay-per-view scheme in which the token will be valid for a certain time period. The time a token is valid for is dependent on what the user selects and what options are available for the selected stream. The user contacts the transaction server, via a client interaction module 300, with the user information and the URI. The transaction server will determine what time options are available for the token and present that to the user. After the user selects the required time limit, the request is passed off to the user verification module 310. Hardware devices suitable for use as a client interaction module within the transaction server include Random Access Memory. Such software is either embedded in the client interaction module or uploaded therein. The software functions to create a process wherein the user information is verified against the database and a valid token is created based upon the options requested by the user.

The user verification module 310 checks for user information passed against a user database to see if the user is valid or not. The user database resides in memory of the user verification module. Hardware devices suitable for use as a user verification module within the transaction server, include Random Access Memory. Such software is either embedded in

the user verification module or uploaded therein. The software functions to create a process wherein the token passed is verified. The URI creation module 320 and the token creation module 330 are tied together and the token is based upon the request URI. This means that the token is unique to the request URI and cannot be used for any other stream. This information is then passed back to the client via module 300. Hardware devices suitable for use as a URI creation module and token creation module, each located within the transaction server, include NA. Such hardware components include software that functions to Random Access Memory. Such software is either embedded in the URI creation module or token creation module or uploaded therein. The software functions to create a process wherein a valid URI to the media stream the user selected is created.

With regard to Figure 4, the client 400 executes and the client is loaded with a URI and a token 410. The client either double clicks on the client's icon (no) or it launched by a media server (yes). If the media server launched the client, there will be a request URI and token in the command-line parameters of the client. A display a window (420) lists all the purchased (and current) data (video) streams available to view. The user will be able to select a data stream to view by double clicking on the title of the stream. The screen waits for input from the user (430) and the user selects a data stream or another housekeeping option (440). If a housekeeping option was selected, execute user request (450) and go back to displaying video streams with module 420.

If the user launches a data stream (selects yes from 410) a URI and token is saved in the purchased video streams list so it can be viewed again at a later time 460. A connection to the streaming server is opened and the URI, token and user information is sent to the streaming server 470. The streaming server acknowledges a valid (or invalid) URI and token combination 480. If the token is invalid or has expired, the server will close the connection and the client will go back and display all the data streams that are available to view. If the server acknowledges a valid URI and token combination, the client will start to receive data from the streaming server and display it 490.

If the data stream finishes or the user selects any of the available stream options such as pause, stop, play, or restart 500, the stream will stop and await further user input. If the stream has finished playing 510, the process goes back to the list of available streams 420, or continue displaying the data stream 490 by processing a user request 520 and then going back to displaying the stream 490.

With regard to Figure 5 and the process run by the streaming server, there is first a connection with the client control module 200, 600 to allow the client to establish a connection with the streaming server. The client will provide the URI, token and user information 610 from user 470. The streaming server determines if the token and URI are valid 620. If the token is invalid or has expired, the connection to the client will be closed with an appropriate error message 630. If token is valid, a set of unique encryption keys will be negotiated with the client 640. A URI will be opened and streaming data will be read into a buffer 650.

The client flow control module 230, provides for the client and streaming server to have a flow control connection established to make sure that the data stream is leaving the streaming server at the same rate it is being used at the client end 660. This addresses bandwidth issues as well as making sure that the client play buffer is not overwritten. Therefore, the client flow control mechanism 660 uses the client flow control module 230 to obtain feedback from the data buffer in the client 710 and control the rate of the data stream to keep the client buffer as full as possible. If the client can not accept any more data at this time, return to flow control module so indicates 670 to slow down or pause the streaming data. If the client can accept more data 680, the client flow control will first determine if there are more data to stream 680. If there are no more data to stream, the data stream could be completed and the client connection will be closed 690. If there is more data to be sent, the data waiting in the send buffer will be encrypted 700 and those data in the send buffer will be sent to the client 710.

With regard to Figure 6 at the transaction server, the client first connects to the transaction server, for example through a web page 800. Preferably, the transaction server will be implemented with ASP scripts. The client sends request URI and user information through ASP command-line arguments 810 and the transaction server user verification module 310 will determine the time limits of available tokens and display to user for selection. The transaction server will look up user information 820 in a database in the user verification module 310. Examples of looking up user information are whether or not a user has an account (exists according to the transaction server) 830. If the user does not have an account 840, a transaction will be opened up to create new account page and get information from the user 840. In addition, the transaction server user verification module 310 will determine if the URI that was requested is free of charge 850. If the URI costs money 860, the transaction server user verification module 310 will debit a credit card that is in the user database. This process will create a URI in the URI creation module 320 of the transaction server.

Once a URI is provided and either paid for or provided free, a token will be created 870 in the token creation module 330. The token now created will be linked with the URI and a time limit will be selected 880. Lastly, the viewer will be started on the client machine and sent back to the client along with the URI and the created token.

We claim:

1. A streaming server for encrypting a data stream to secure the data stream to enable only single viewing, comprising:

(a) a streaming server component, wherein the streaming server component comprises a client data connection module to send data packets to a client, and encryption module to use encryption keys negotiated with the client to encrypt the data stream and operably connected to the client data connection module, and a flow control module for controlling the rate of data stream flow to maintain a full client buffer; and

(b) a transaction server component, wherein the transaction server component comprises a client interaction module for connecting a user to the transaction server component, a user verification module having a user database wherein the user verification module is operably linked to the client interaction module and checking for a valid user, and a URI and token creation module operably linked to the user verification module for creating new URIs and tokens in response to user requests.

2. The streaming server of claim 1, wherein the streaming server component further comprises a read buffer module operable connected with the flow control module for reading in data from a source footage on storage medium.

3. The streaming server of claim 1, wherein the streaming server component further comprises a user interface module operably connected to the file system module or flow control module for setting server options.

4. The streaming server of claim 1, wherein the streaming server further comprises a client server component comprising a data stream control protocol module to create an initial connection to the streaming server component, a decryption module to decrypt the incoming data stream, an input buffer module to buffer incoming data streams, and a display control module to control the display of streaming data.

5. The streaming server of claim 4, wherein the client server component further comprises a display module to display audio and video data.

6. A process for encrypting a data stream to secure the data stream to enable only single viewing, comprising:

(a) providing a client selection for a streaming data transmission;

(b) opening a connection to a streaming server and sending URI, token and user information to the streaming server, wherein the streaming server comprises a client data connection module to send data packets to a client, and encryption module to use encryption keys negotiated with the client to encrypt the data stream and operably connected to the client data connection module, and a flow control module for controlling the rate of data stream flow to maintain a full client buffer;

(c) approving or disapproving a valid or invalid, respectively, URI and token combination on a transaction server, wherein the transaction server comprises a client interaction module for connecting a user to the transaction server component, a user

verification module having a user database wherein the user verification module is operably linked to the client interaction module and checking for a valid user, and a URI and token creation module operably linked to the user verification module for creating new URIs and tokens in response to user requests; and

(d) providing a continuously encrypted data stream to the client if a valid URI and token combination was found.

7. The process of claim 6 wherein the streaming server further comprises a read buffer module operable connected with the flow control module for reading in data from a source footage on storage medium.

8. The process of claim 6 wherein the streaming server further comprises a user interface module operably connected to the file system module or flow control module for setting server options.

9. The process of claim 6 wherein the streaming server further comprises a client server component comprising a data stream control protocol module to create an initial connection to the streaming server component, a decryption module to decrypt the incoming data stream, an input buffer module to buffer incoming data streams, and a display control module to control the display of streaming data.

10. The process of claim 9 wherein the client server component further comprises a display module to display audio and video data.

11. The process of claim 6 wherein the providing the continuously encrypted data stream step (d) further comprises a user interface module in the streaming server to allow for pausing, stopping, playing or restarting the data stream.

12. The process of claim 6 wherein the transaction server is implemented with ASP scripts for encryption.

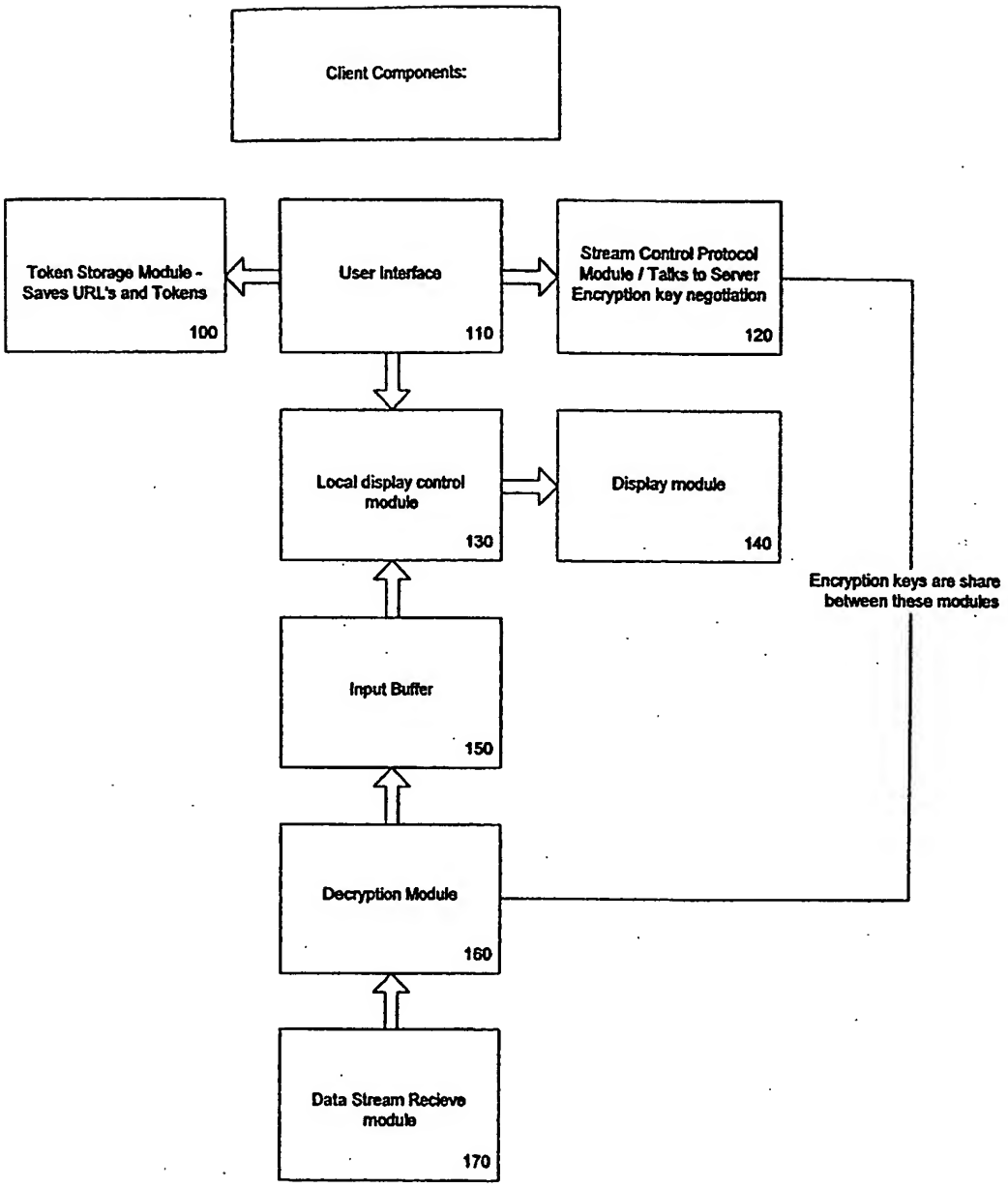

Client Components:
Token Storage Module - Saves URL's and Tokens
100
User Interface
110
Stream Control Protocol Module / Talks to Server Encryption key negotiation
120
Local display control module
130
Display module
140
Encryption keys are share between these modules
Input Buffer
150
Decryption Module
160
Data Stream Recieve module
170

Fig. 1.

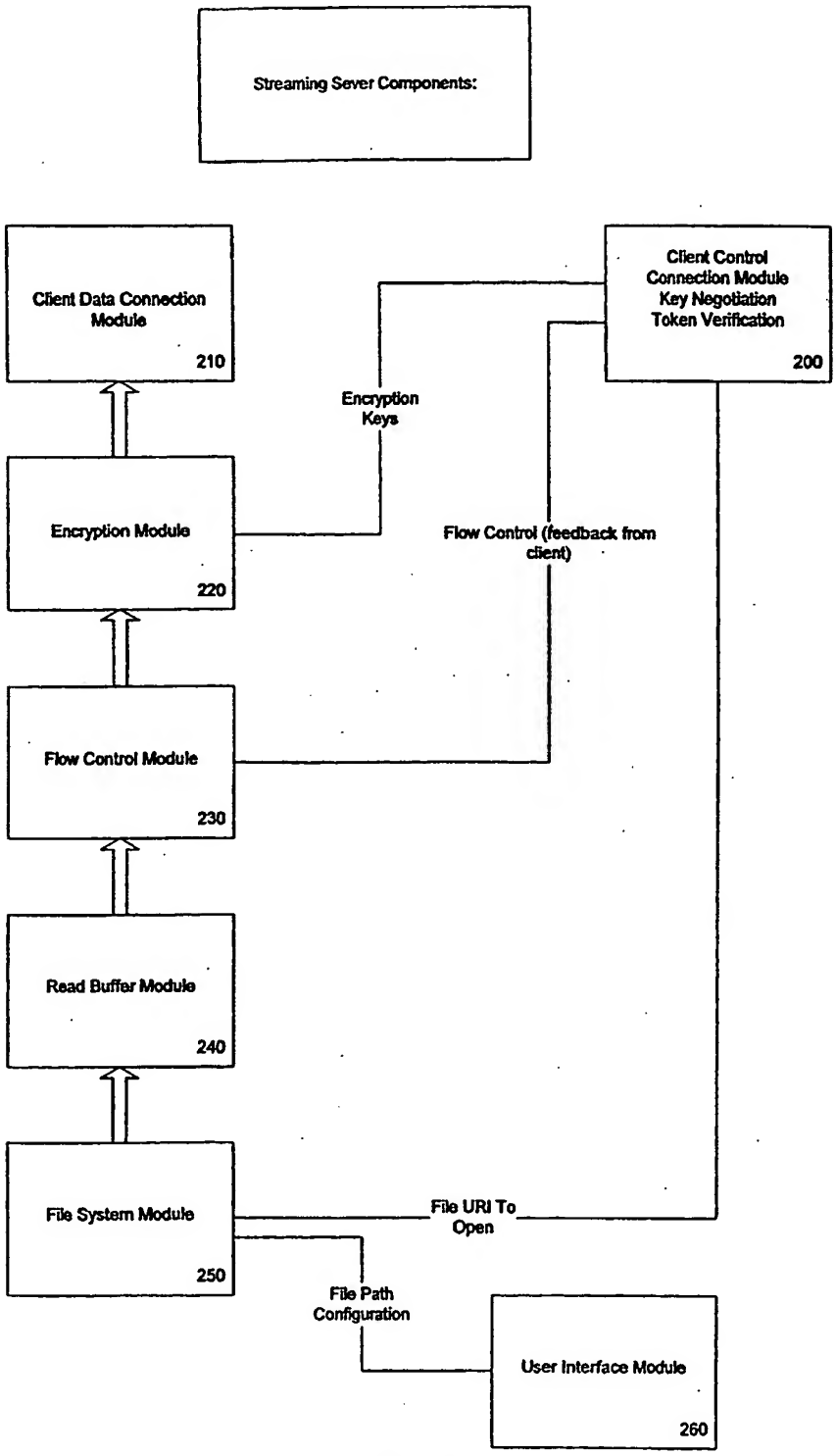

Streaming Sever Components:
Client Data Connection Module
210
Client Control Connection Module
Key Negotiation
Token Verification
200
Encryption Keys
Encryption Module
220
Flow Control (feedback from client)
Flow Control Module
230
Read Buffer Module
240
File System Module
250
File URI To Open
File Path Configuration
User Interface Module
260

Fig. 2

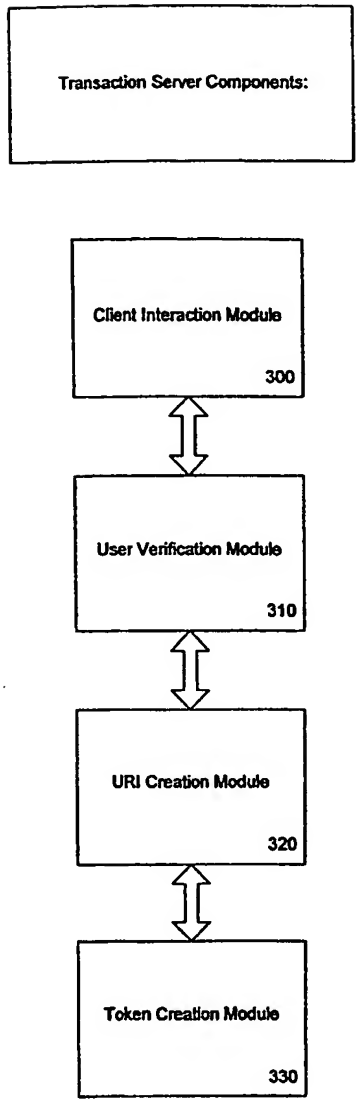

Transaction Server Components:
Client Interaction Module
300
User Verification Module
310
URI Creation Module
320
Token Creation Module
330

Fig. 3

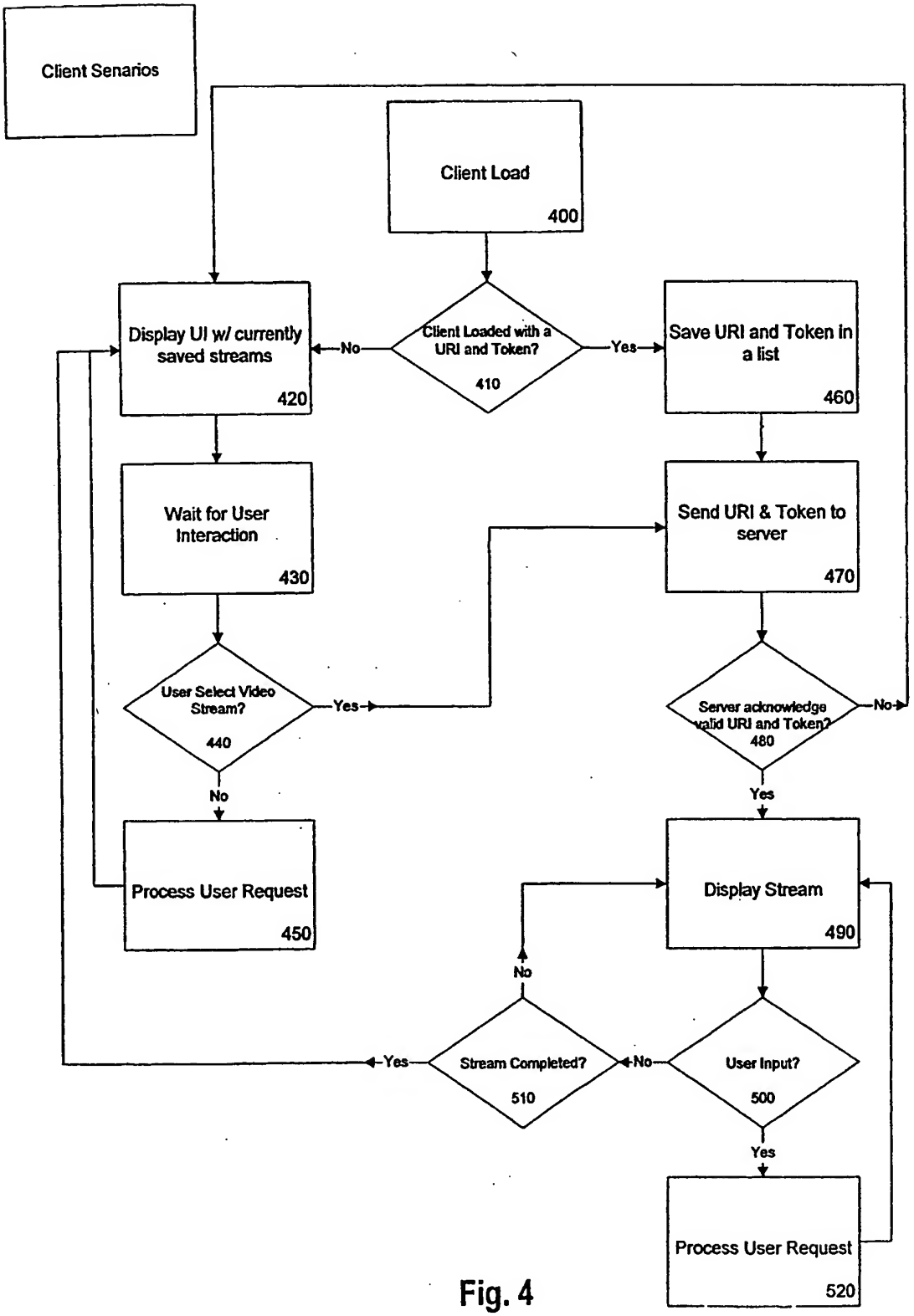

Client Senarios
Client Load
400
Client Loaded with a URI and Token?
410
No
Yes
Display UI w/ currently saved streams
420
Save URI and Token in a list
460
Wait for User Interaction
430
Send URI & Token to server
470
User Select Video Stream?
440
Yes
Server acknowledge valid URI and Token?
480
No
No
Yes
Process User Request
450
Display Stream
490
No
Stream Completed?
510
Yes
User Input?
500
No
Yes
Process User Request
520

Fig. 4

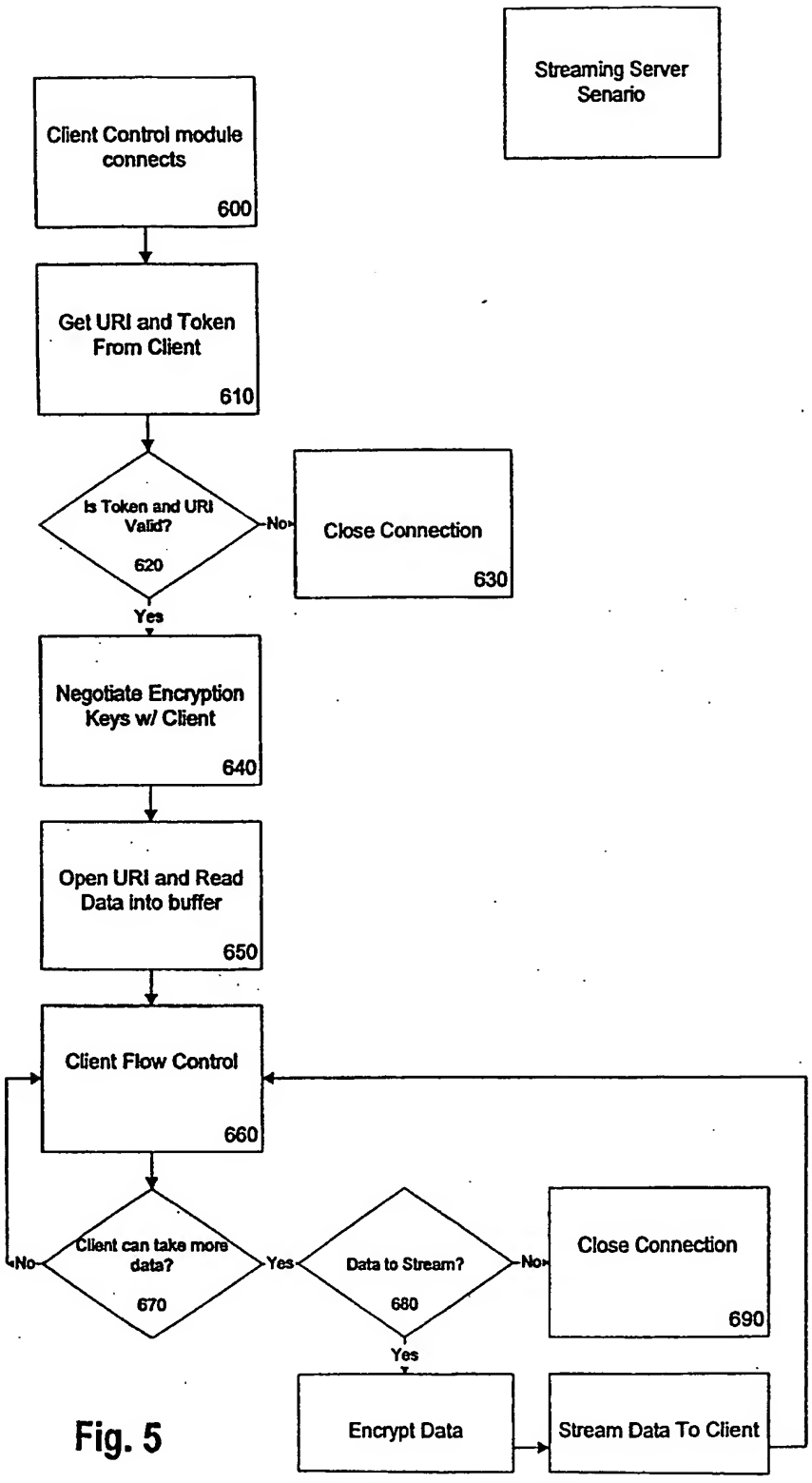

Streaming Server Senario
Client Control module connects
600
Get URI and Token From Client
610
Is Token and URI Valid?
620
No
Close Connection
630
Yes
Negotiate Encryption Keys w/ Client
640
Open URI and Read Data into buffer
650
Client Flow Control
660
Client can take more data?
670
No
Yes
Data to Stream?
680
No
Close Connection
690
Yes
Encrypt Data
Stream Data To Client

Fig. 5

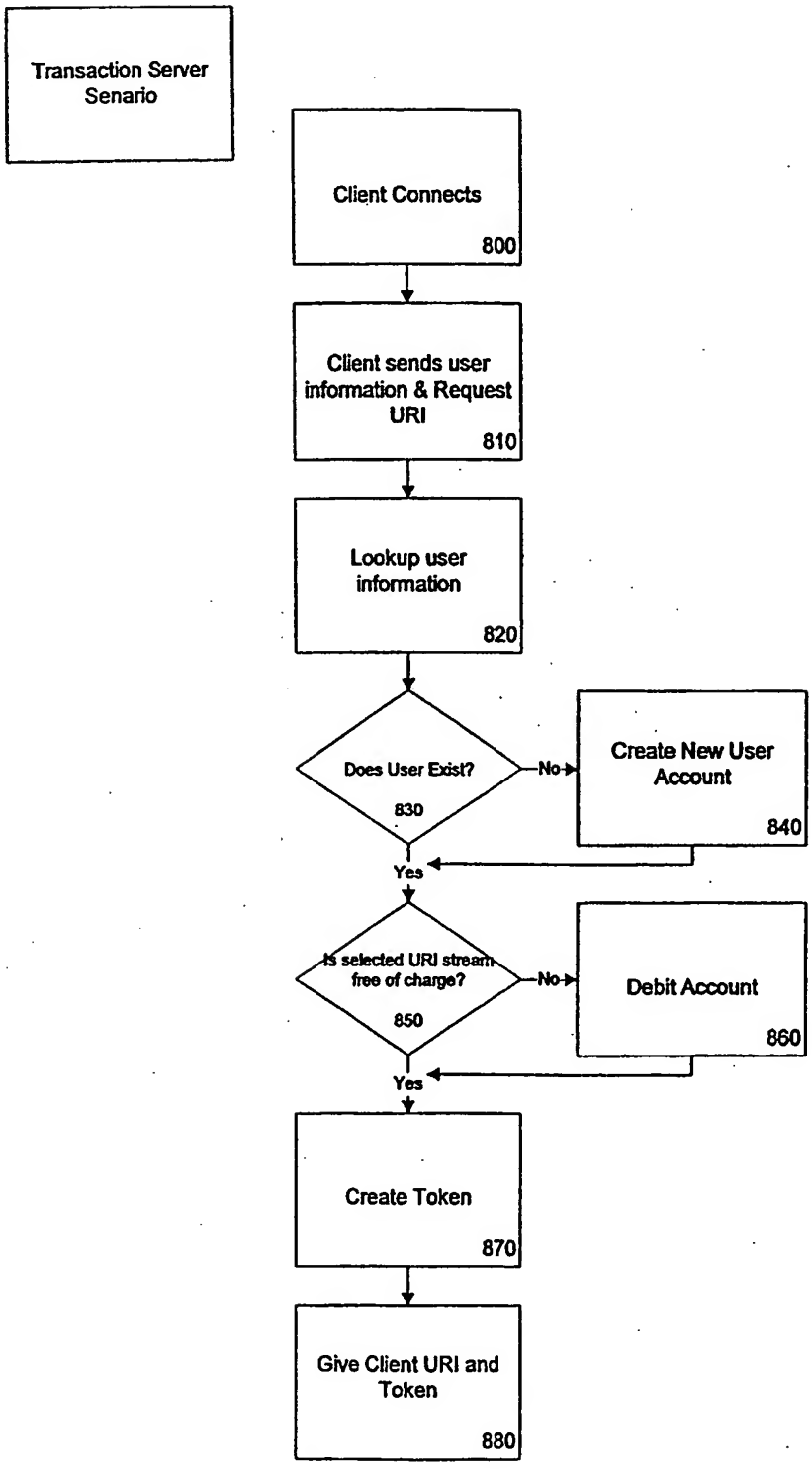

Transaction Server Senario
Client Connects
800
Client sends user information & Request URI
810
Lookup user information
820
Does User Exist?
830
No
Create New User Account
840
Yes
Is selected URI stream free of charge?
850
No
Debit Account
860
Yes
Create Token
870
Give Client URI and Token
880

Fig. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/30899

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :H04L 9/00
US CL :380/200, 201; 713/150, 168

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 380/200, 201; 713/150, 168

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WEST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,953,005 A (LIU) 14 September 1999, col. 1, lines 6-9; col.2, lines 40-45; col. 4. line 63 to col. 5, line 49.	1-12
Y	US 5,883,957 A (MOLINE et al) 16 March 1999, col. 11, line 22 to col. 12, line 11; col. 12, lines 33-46; col. 19, line 34 to col. 21, line 12.	1-12
Y	US 5,825,879 A (DAVIS) 20 October 1998, col. 3, lines 54-61; col.4, lines 3-14; col. 4, lines 49-55; col. 6, lines 25-41.	1-12

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 28 FEBRUARY 2001	Date of mailing of the international search report 12 MAR 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer MATTHEW SMITHERS Telephone No. (703) 308-9293

Form PCT/ISA/210 (second sheet) (July 1998)*